

Darshan University

A Project Report on

**“Hostel Management System”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – VI

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Gohel Krish V. | Enrollment No.: 22010101060 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof.  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2101CS503)** for entitled **“Hostel Managment System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of (**Guide name)** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Gohel Krish V.

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Hostel Management System” has** been satisfactorily prepared by **Gohel Krish V.**(**22010101060**) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2023-2024.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof., Guide Name  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. **Guide name** and all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Gohel Krish V.**

**ABSTRACT**

Hostel management system is a system which aims in developing a computerized system to maintain all the daily work of hostel. This system will act as a tool to managing records of hostel, rooms, beds, students, student registration in database instead of Management Books. This project has a facility of admin login through which the admin can monitor the whole system.  This system will be designed with the basic features such as hostel can add/view/update/delete student’s details, their allotment of room, bed ,mess, details, maintain attendance of students, maintain bill of mess, rooms in it. It has also a facility where student after logging in their accounts can see their allotted room, bed no. from starting date and return date and also they can request the hostel to add required facility in room by filling the room-query request form. The Hostel after logging into his account i.e. admin account can generate various reports such as student report , room report , bed report and students attendance report.

Main purpose of this system is to reduce human efforts as much as possible.

**Table of Contents**

[List of Figures I](#_Toc146706805)

[List of Tables II](#_Toc146706806)

[1 Introduction 1](#_Toc146706807)

[1.1 Product perspective 1](#_Toc146706808)

[1.2 Product features 1](#_Toc146706809)

[1.2.1 There are three different users who will be using this product: 1](#_Toc146706810)

[1.2.2 The features that are required for the Librarian are: 1](#_Toc146706811)

[1.2.3 The features that are required for the Member are: 1](#_Toc146706812)

[1.3 Functional Requirement 2](#_Toc146706813)

[1.3.1 Librarian **Error! Bookmark not defined.**](#_Toc146706814)

[1.3.2 Member 2](#_Toc146706815)

[1.4 Non-Functional Requirement 2](#_Toc146706816)

[1.4.1 Usability: 2](#_Toc146706817)

[1.4.2 Accuracy: 2](#_Toc146706818)

[1.4.3 Availability: 3](#_Toc146706819)

[1.4.4 Maintainability: 3](#_Toc146706820)

[2 Design and Implementation Constraints 4](#_Toc146706821)

[2.1 Use case diagram 4](#_Toc146706822)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc146706823)

[2.3 Sequence diagram 9](#_Toc146706824)

[2.4 State diagram 10](#_Toc146706825)

[2.5 Class diagram 11](#_Toc146706826)

[2.6 Data flow diagram 12](#_Toc146706827)

[2.6.1 Context diagram (level-0) 12](#_Toc146706828)

[2.6.2 DFD Level-1 12](#_Toc146706829)

[2.6.3 DFD Level-2 13](#_Toc146706830)

[3 External interface requirement (Screens) 14](#_Toc146706831)

[3.1 Screen-1: Registration Form 14](#_Toc146706832)

[3.2 Screen-2: Login Form 15](#_Toc146706833)

[3.3 Screen-3: Add borrower book 16](#_Toc146706834)

[4 Database design 17](#_Toc146706835)

[4.1 List of Tables 17](#_Toc146706836)

[5 Stories and Scenario 18](#_Toc146706837)

[5.1 Story-1: Add New Book in Library Catalogue 18](#_Toc146706838)

[5.1.1 Scenario# S1.1 18](#_Toc146706839)

[5.1.2 Scenario# S1.2 18](#_Toc146706840)

[5.1.3 Scenario# S1.3 18](#_Toc146706841)

[5.2 Story-2: Search Book 19](#_Toc146706842)

[5.3 Story-3: Manage due date for borrowed book 19](#_Toc146706843)

[5.4 Story-4: Renew book 19](#_Toc146706844)

[5.5 Story-5: Generate a report on book usage and availability 19](#_Toc146706845)

[6 Test cases 20](#_Toc146706846)

[7 References 24](#_Toc146706847)

# List of Figures

[Figure 2.1‑1 Use case diagram for library management system 4](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Member Registration 5](#_Toc146706849)

Figure 2.2-2 Activity diagram for Disciplinary Issues……………………………………………………………………………………..6

[Figure 2.2‑3 Swimlane diagram for Member Registration 7](#_Toc146706850)

Figure 2.2-4 Swimlane diagram for Exit Apply……………………………………………………………………………………………….8

[Figure 2.3‑1 Sequence diagram for Book Issue 9](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 10](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 10](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 11](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 12](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 12](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 13](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 14](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 15](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 16](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 14](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 15](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 16](#_Toc146706863)

[Table 4.1‑1 Table: Book 17](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 17](#_Toc146706865)

[Table 4.1‑3 Table: Staff 17](#_Toc146706866)

[Table 4.1‑4 Table: Student 17](#_Toc146706867)

# Introduction

## Product perspective

This project is basically updating the manual Hostel system into an internet-based application so that the users can know the details of their account, allocated room and bed, details about mess and room bills, etc. It is a multi-user version and can take care of all the fundamental functions of a hostel like records of students, room bill ,mess bill, etc. It can satisfactorily cater to all the basic functions of a hostel.

## Product features

### There are three different users who will be using this product:

* Hostel Administrator who will be acting as the administrator.
* Member who will be check their details.
* Guest who will request for membership..

### The features that are required for the Hostel Administrator are:

* control the administrator to log in using secure credentials.
* Search if you allocate the room to the specific student in your hostel based on room number, hostel id etc.
* Generate the hostel id for the student.
* track and log entry and exit times for students and visitors.
* Can add, update, and delete students records.
* Can allocate a room to the student.
* Can view the list of rooms available in each category.
* Administrator can record and track payment details from students.
* provide options for managing and tracking hostel fees and other charges.
* Add room information to the database.
* Edit the information of the existing room.
* Can access all the accounts of the students.
* send announcements and notifications to students.
* support automated email or SMS notifications for important events (e.g., payment due dates, maintenance schedules).
* manage access control for different areas of the hostel.
* submit complaints or issues through a digital portal.
* manage room availability and assignments.
* manage staff schedules and attendance.

### The features that are required for the Member are:

* register and create a profile with personal details.
* Can update their profile information
* display the member's room assignment details.
* members request room changes.
* allow members to view their payment history.
* enable members to pay hostel fees & mess fees online.
* allow members to log visitor details and expected visit times.
* allow members to submit feedback or suggestions regarding hostel services.
* allow members to submit complaints or issues online.

## Functional Requirement

### Student/Hosteler

* Login : Students should be able to login themselves into the hostel management system and manage their profiles.
* Room Allocation: The system should facilitate the allocation and assignment of rooms to students.
* Fees Payment: Students should be able to view their hostel fees, make payments, and receive invoices electronically.
* Maintenance Request And Tracking: Student should be able to submit maintenance requests and track their status
* Community Notice: Students should have access to hostel announcements, notices, and community events.
* Complaints : Students should be able to submit complaints and suggestions through the system.
* Feedback: Students should be able to submit feedback through the system.
* Security And Access Control: The system should ensure secure access to hostel facilities and personal information.
* Calendar And Event Booking: Students should be able to view and book hostel facilities for events or activities.
* Health And Wellness Resources: Students should have access to health and wellness resources and information through the system.
* Transportation Service Information: Students should be able to access information about transportation services available near the hostel.
* Notification Preferences and Alerts: Students should be able to customize their notification preferences and receive alerts related to hostel activities.
* Lost And Found: Students should have a mechanism to report lost items and track their status.

### Parents

* Parent Registration and Access: Parents should be able to register themselves and obtain access to the hostel management system.
* Student Information and Profile Viewing: Parents should be able to view their child's hostel profile and personal information.
* Room Allocation and Roommate Information: Parents should have access to information about their child's room allocation and roommate.
* Fee Payment and Financial Transactions: Parents should be able to view hostel fees, make payments, and receive financial statements.
* Emergency Contact and Notification Settings: Parents should be able to update emergency contact information and set notification preferences.
* Health and Medical Records Access: Parents should have access to their child's health and medical records stored in the hostel management system.
* Communication with Hostel Staff: Parents should be able to communicate with hostel administrators or staff members.
* Events and Activities Participation: Parents should be informed about hostel events, activities, and participation opportunities.
* Safety and Security Measures: Parents should have access to information about hostel safety and security measures.
* Educational Support and Resources: Parents should have access to educational resources and support services available through the hostel.
* Community Engagement and Updates: Parents should receive updates and newsletters about hostel community news and developments.
* Transportation and Travel Information: Parents should have access to information about transportation options and travel arrangements near the hostel.

### Warden

* Dashboard and Overview: Admin should have secure login credentials and role-based access to the system.
* Room Allocation: Ability to assign rooms, update allocations, and handle room transfers.
* Students Registration and Records: Capture student details (name, contact, course) and maintain records.
* Billing and Payments: Generate bills, track payments, and manage financial transactions.
* Attendance Tracking: Record student check-ins, check-outs, and attendance.
* Complaint Manage: Receive, assign and track complaints from students.
* Inventory Manage: Maintain Stock of hostel supplies(bedding, toiletries etc.)
* Maintenance Requests: Handle maintenance requests(plumbing, electrical etc.)
* Housekeeping schedules: Set Cleaning Schedules For Rooms And Common Areas.
* Security And Access Control: Manage access cards, CCTV monitoring, and security protocols.
* Emergency Procedure: Define evacuation plans, emergency contacts, and protocols.
* Event Manage: Plan And Organize Hostel events (workshops, celebration etc)
* Document Manage: Store And Manage Important Documents(contracts, policies).
* System Customization And Setting: Configure System Parameters(tax rates, room types).

### Admin

* User Manage: Administrators should be able to manage user accounts and permissions.
* Room Allocation and Management: Administrators should manage room allocation, assignments, and occupancy.
* Fees: Administrators should manage hostel fees, payment schedules, and financial transactions.
* Reporting and Analytics: Administrators should access reports and analytics on hostel occupancy, financial performance, and operational metrics.
* Complaints and Grievance Handling: Administrators should handle complaints, feedback, and grievances from students and parents.
* Security and Access Control: Administrators should manage security measures and access control within the hostel premises.
* Event and Facility Booking: Administrators should manage bookings for hostel facilities and coordinate events.
* Health and Safety Compliance: Administrators should ensure compliance with health and safety regulations within the hostel premises.
* Training and Staff Development: Administrators should manage training programs and professional development for hostel staff.
* Transportation Services Coordination: Administrators should coordinate transportation services for students living in the hostel.
* Document Management and Archiving: Administrators should manage documentation and archives related to hostel operations.
* Quality Assurance and Continuous Improvement: Administrators should implement quality assurance measures and initiatives for hostel services.
* Scalability and Customization: The HMS should be scalable and customizable to accommodate growth and meet specific institutional requirements.

## Non-Functional Requirement

* + 1. Usability:
* The user interface shall be intuitive and user-friendly, requiring minimal training for hostel staff to use effectively.
  + 1. Scalability:
* It should be capable of handling an increase in the number of hostels and students without requiring major architectural changes.
  + 1. Availability:
* The system should be available 99.99% of the time to ensure uninterrupted access for users.
  + 1. Reliability:
* Data integrity should be maintained with a backup and recovery mechanism to ensure minimal data loss in case of system failure.
  + 1. Accessibility:
* The system should be accessible from different devices (desktops, tablets, mobile phones) and compatible with popular web browsers.
  + 1. Localization:
* It should support multiple languages and currencies to accommodate international students and staff.
  + 1. **Response Time**:
* The average response time for user actions (e.g., booking rooms, updating records) should be less than 2 seconds.
  + 1. **Backup and Recovery**:
* Regular automated backups of data should be performed with a tested recovery procedure to ensure quick restoration of services in case of data loss.
  + 1. User Support:
* There should be a dedicated helpdesk or support team to assist users with system-related queries and issues.
  + 1. **Backup and Recovery**:
* Regular automated backups of data should be performed with a tested recovery procedure to ensure quick restoration of services in case of data loss.

# Design and Implementation Constraints

## Use case diagram

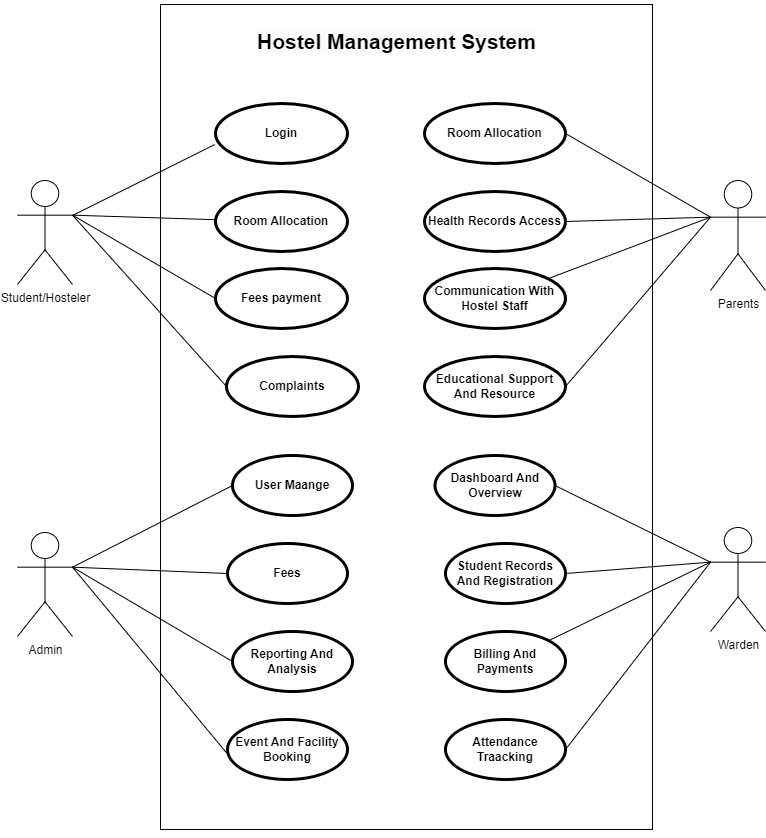


Figure 2.1‑1 Use case diagram for hostel management system

## Activity diagram and Swimlane diagram

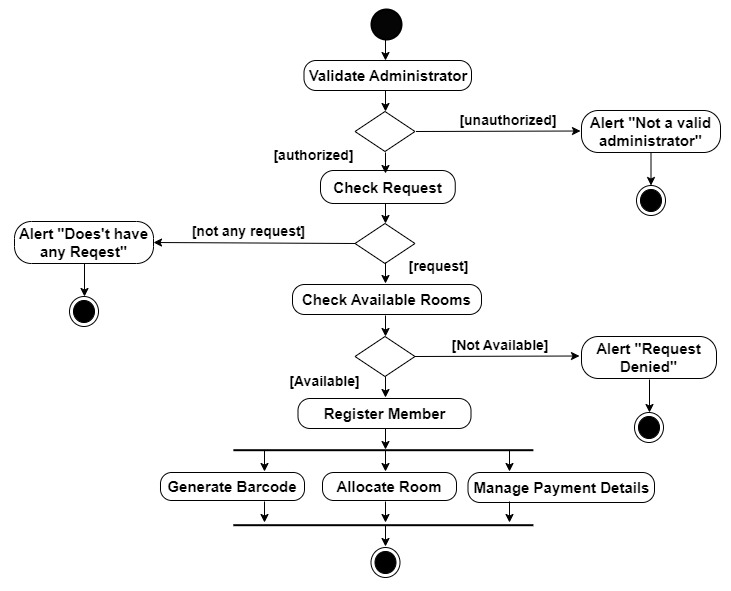


Figure 2.2‑1 Activity diagram for Member Registration

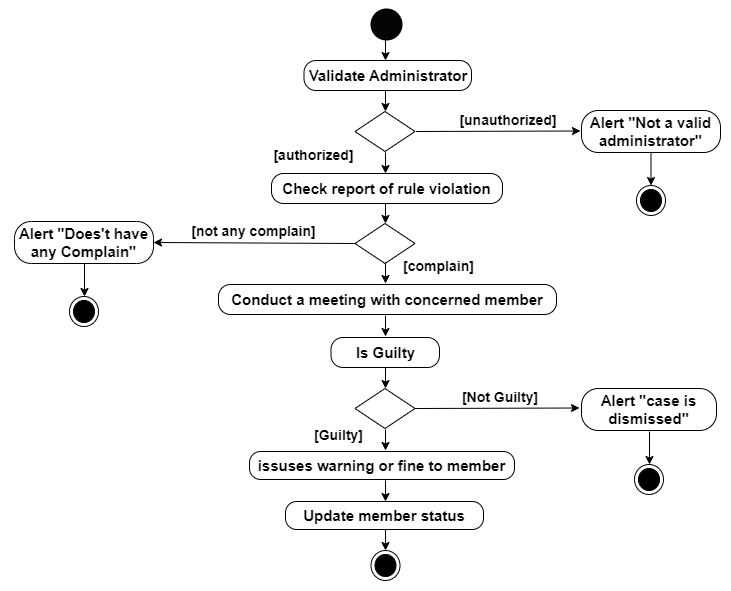


Figure 2.2‑2 Activity diagram for Member Disciplinary Action

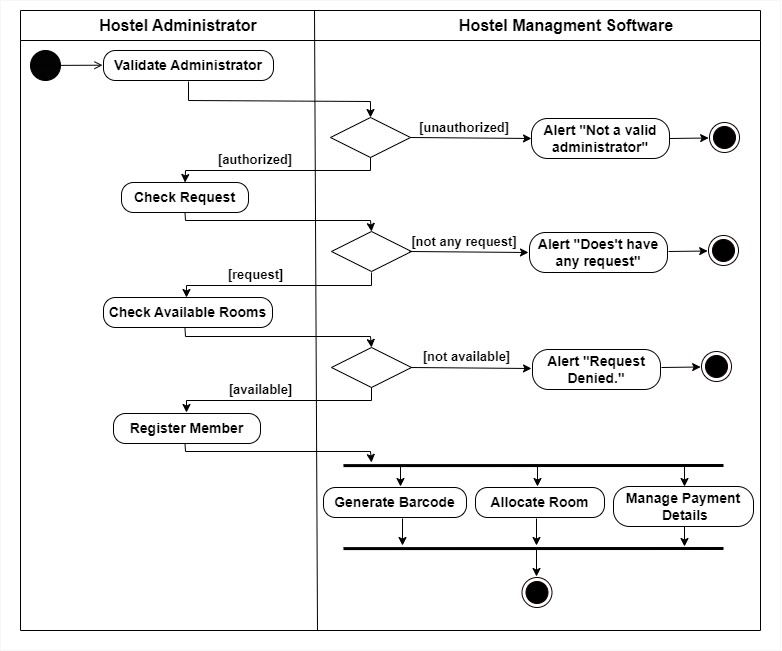


Figure 2.2‑3 Swimlane diagram for Member Registration

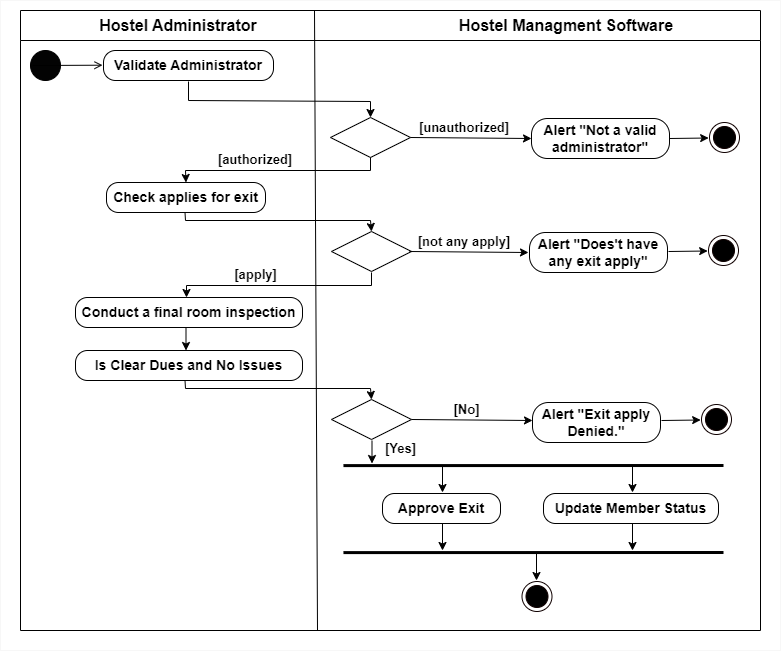


Figure 2.2‑4 Swimlane diagram for Member exit apply process

## Sequence diagram

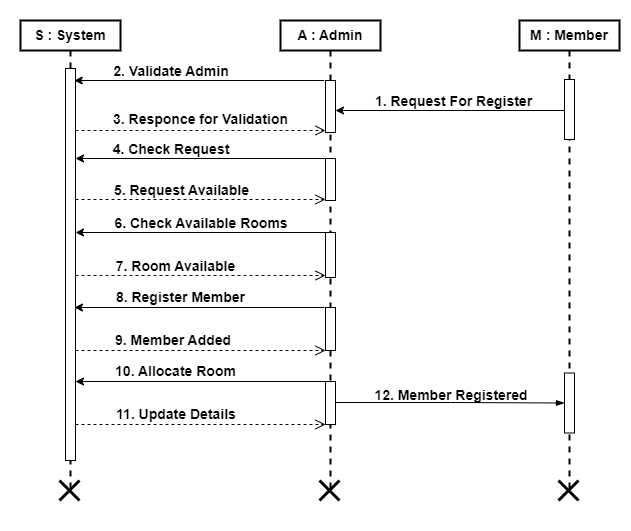


Figure 2.3‑1 Sequence diagram for Member Registration

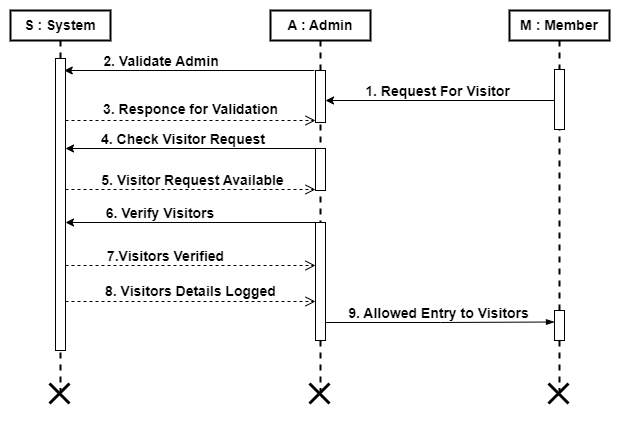


Figure 2.3‑1 Sequence diagram for Visitor Management

## State diagram

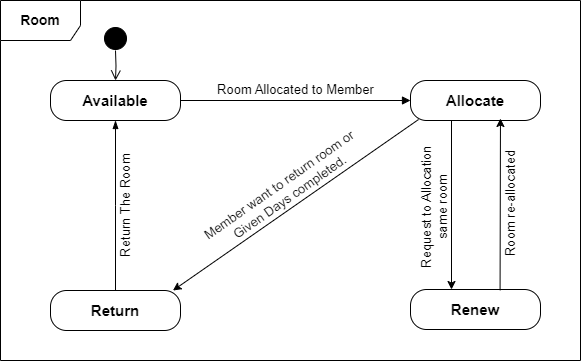


Figure 2.4‑1 State diagram of Room

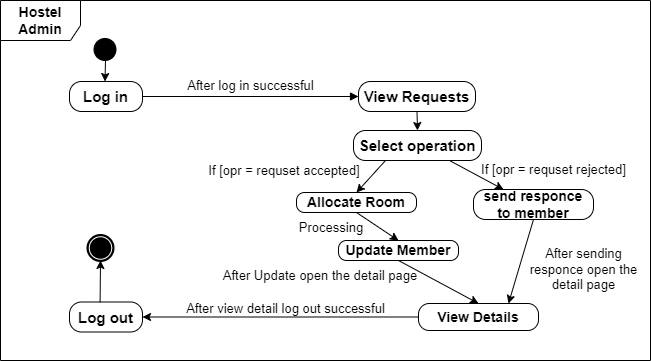
`

Figure 2.4‑2 State diagram for Hostel Admin

## Class diagram

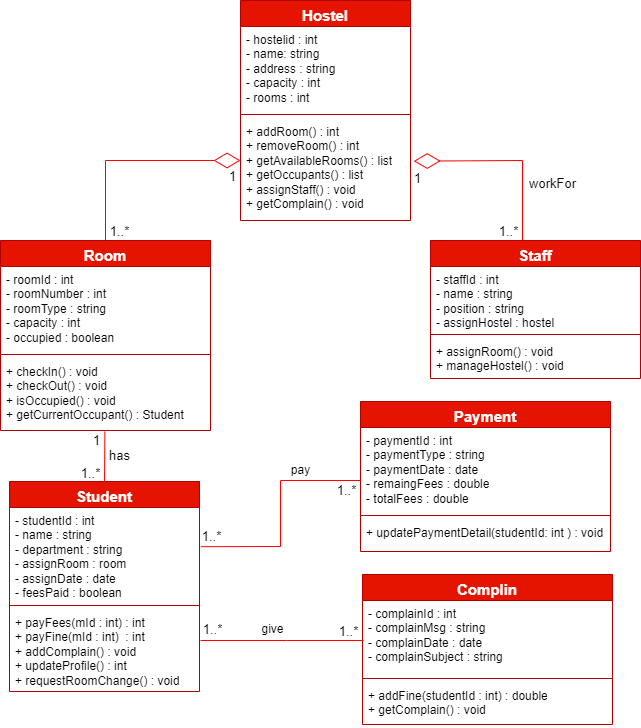


Figure 2.5‑1 Class diagram for Hostel management system

## Data flow diagram

### Context diagram (level-0)

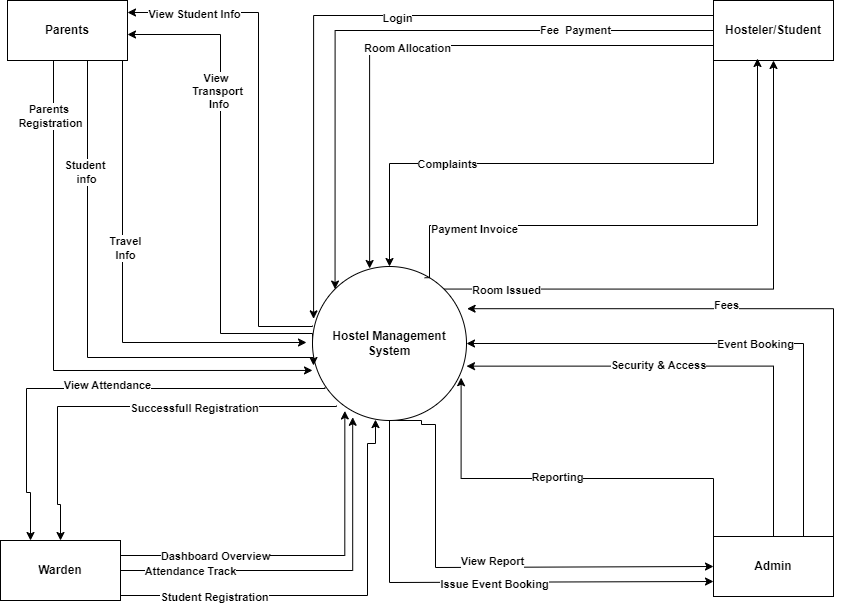


Figure 2.6‑1 Context diagram for Hostel management system

### DFD Level-1

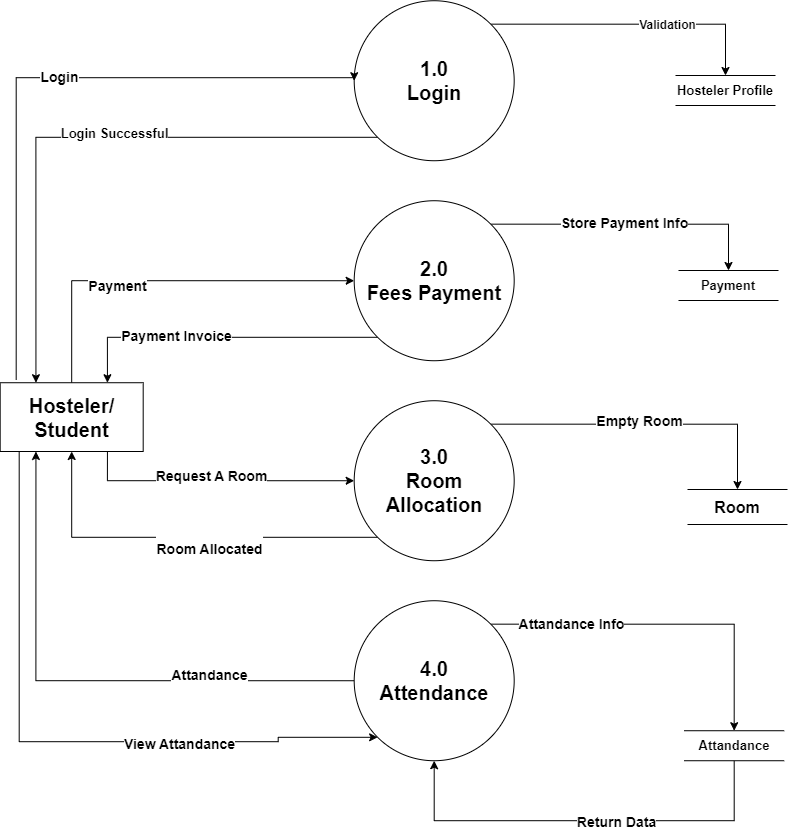


Figure 2.6‑2 DFD level-1 for Hostel management system

### DFD Level-2

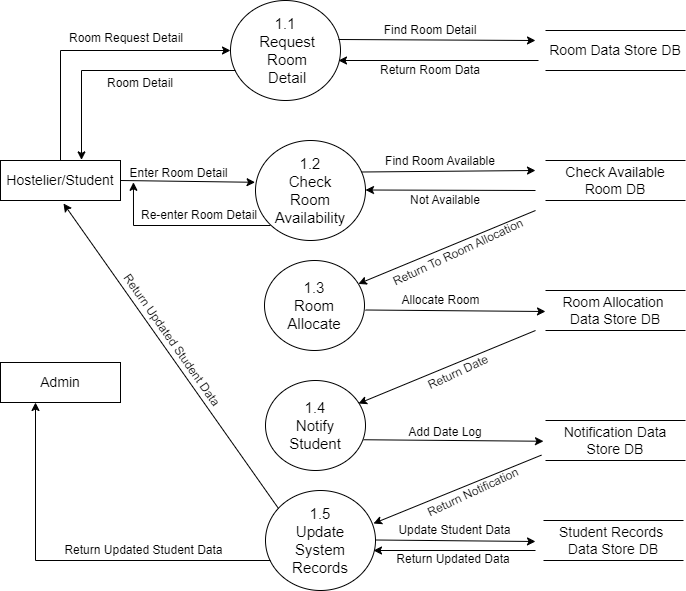


Figure 2.6‑3 DFD level-2 for Room Allocation

# External interface requirement (Screens)

## Screen-1: Registration Form

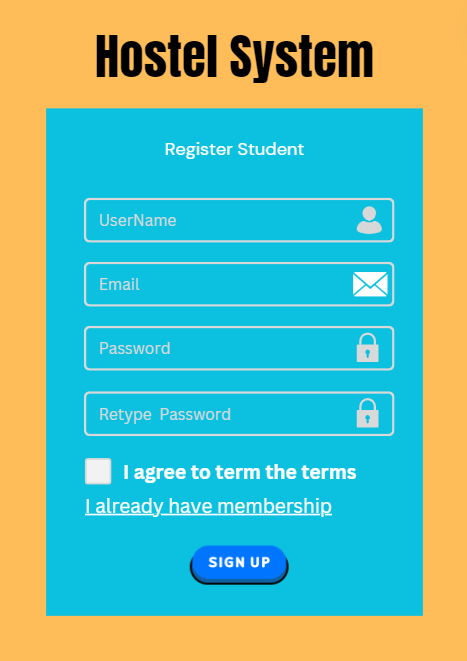


Figure 3.1‑1 Screen-1: Registration Form

**Purpose:** This form will allow the target end-users to register in the system. To register , the following information will be encoded in the system.

Table 3.1‑1 Screen element of Registration form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 4 | Retype password | Password | M | 1 | Retype Password field should be editable and accept the password and display as star or dot. |
| 5 | I agree to the terms | Checkbox | M | 1 | Checkbox for accepting terms and condition. |
| 6 | I already have a membership | Link | ------ | ------ | Link for navigate to login page for registered user. |
| 7 | Register | Button | ------ | ------ | Register is a button for store the entered data into database. |

## Screen-2: Login Form

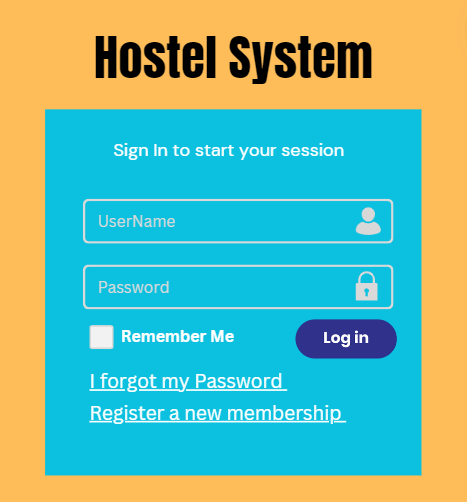


Figure 3.2‑1 Screen-2: Login Form

**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

Table 3.2‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 3 | Remember Me | Checkbox | M | 1 | Saving login credentials through remember me checkbox |
| 4 | I forgot my password | Link | ------ | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| 5 | Register a new membership | Link | ------ | ------ | Link for navigate to membership registration. |
| 6 | Sign in | Button | ------ | ------ | Login button navigates to another page even if valid login credentials. |

## Screen-3: List All Product Page

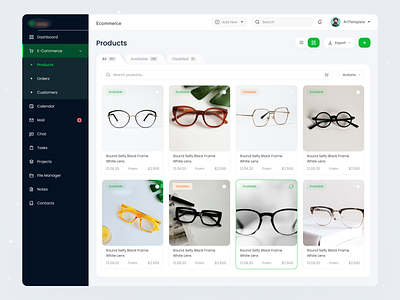


Figure 3.3‑1 Screen-3: List All Product Page

**Purpose:** It helps to find Product according to user’s need.

Table 3.3‑1 Screen element of List All Product Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Sidebar | Links | M | N | Navigate between Pages. |
| **2** | Title of Webpage | Links | M | 1 | Navigate to Home Page |
| **3** | Add new | Button | M | 1 | To Add New Product. |
| **4** | Search bar | Input | M | 1 | To search the product. |
| **5** | Seller Profile | Button | O | 1 | Details of Logged in Seller. |
| **6** | Products title | Text | M | 1 | Title Product. |
| **7** | View Type | Button | M | N | To change view Of all products. |
| **8** | Filtered Product | Button | O | 1 | To See product according to filters. |
| **9** | Product Image | Image | M | N | Image of the Product. |
| **10** | Availability status | Label | M | N | Product is available or not. |
| **11** | Product Details | Text | M | N | Details of the Product. |
| **12** | Product size | Text | M | N | It give various options in size. |
| **13** | Product Price | Text | M | N | Price of the Product. |

## Screen-4: Product Detail Page

A comb with text and images

Description automatically generated

Figure 3.4‑1 Screen-3: Product Detail Page

**Purpose:** This page helps user to explore whole product in detail.

Table 3.4‑1 Screen element of Product Detail Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Offer details | Text | O | 1 | Details of product offer if offer exists. |
| **2** | Navbar | Links | M | N | Navigate between Pages. |
| **3** | Brand Name | H2 | M | 1 | Name of the brand. |
| **4** | Search bar | Input | M | 1 | To search the product. |
| **5** | User Profile | Button | O | 1 | Details of Logged in user. |
| **6** | Cart | Button | M | 1 | To Navigate the cart page where all product of that user which is added to cart. |
| **7** | Product Image | Image | M | 1 | Image of the product. |
| **8** | Next Image Button | Button | O | 1 | To See Other Image of that product. |
| **9** | Product Name | Text | M | 1 | Name of the Product. |
| **10** | Product Image | Paragraph | M | 1 | Detailed use and explanation of product. |
| **11** | Quantity Button | Button | M | 1 | It helps to increase or decrease quantity of the product. |
| **12** | Add to Cart Button | Button | M | 1 | It helps user to add product in cart to buy letter. |

## Screen-5: E-Commerce Home Page

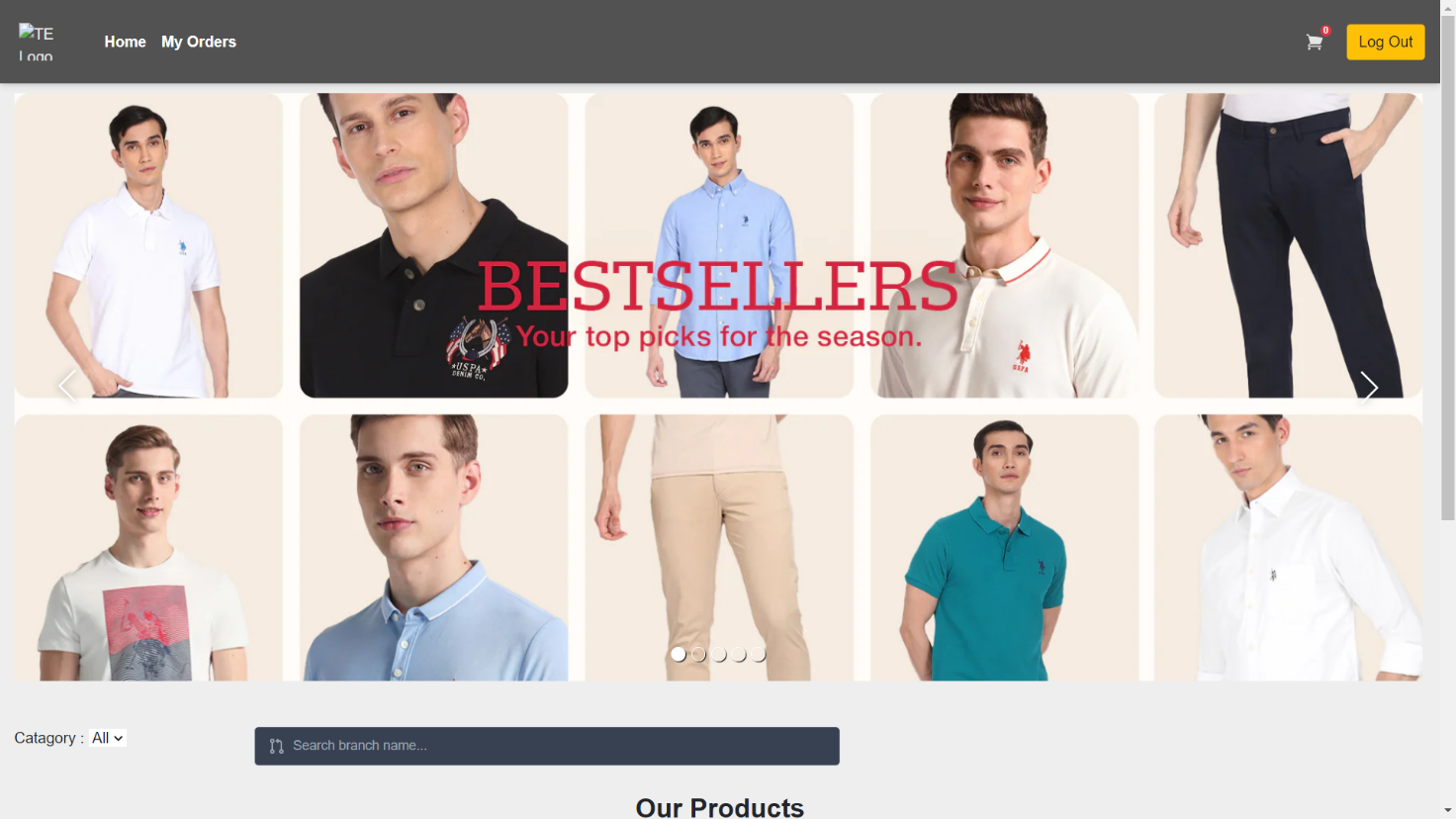


Figure 3.5‑1 Screen-3: E-Commerce Web Home Page

**Purpose:** This page helps user to explore different Product with it’s list.

Table 3.5‑1 Screen element of E-Commerce Web Home Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| 1 | Logout | Button | M | 1 | Log Out Screen |
| 2 | Navbar | Link | M | N | To Display Menu |
| 3 | Logo | Image | M | 1 | Showing To Web Logo |
| 4 | Cart | Button | M | 1 | Display Add Product Count |
| 5 | Search Bar | Textbox | M | 1 | Search Product Name |
| 6 | Category | Dropdown | M | 1 | Display All Category |
| 7 | Image | Image | O | N | Display Product Image |

# Database design

## List of Tables

* Admin
* Warden
* Hosteler
* Parents
* Room

Table 4.1‑1 Table: Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| AdminID | int | NN | Primary Key | Auto Increment Start With 1 |
| UserName | varchar(100) | NN | - | - |
| Password | varchar(25) | NN |  | - |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |
| HostelerID | Int | NN | Foreign Key | Reference Of Hosteler Table |
| ParentID | Int | NN | Foreign Key | Reference Of Parents Table |
| WardenID | Int | NN | Foreign Key | Reference Of Warden Table |

Table 4.1‑2 Table: Hosteler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| HostelerID | int | NN | Primary Key | Auto Increment Start With 1 |
| FullName | varchar(100) | NN | - | - |
| Gender | varchar(10) | NN | - | - |
| DateOfBirth | Datetime | NN | - | - |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |
| AdmissionDate | Datetime | NN | - | - |
| RoomID | Int | NN | Foreign Key | Reference Of Room Table |

Table 4.1‑3 Table: Parents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| ParentID | int | NN | Primary Key | Auto Increment Start With 1 |
| FulllName | varchar(100) | NN | - | - |
| RelationShip | Varchar(50) | NN | - | Relationship to the hosteler (e.g., Father, Mother, Guardian) |
| HostelerID | int | NN | Foreign Key | Reference Of Hosteler Table |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |

Table 4.1‑4 Table: Warden

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| WardenID | int | NN | Primary Key | Auto Increment Start With 1 |
| FullName | varchar(100) | NN | - | - |
| Gender | varchar(100) | NA | - | - |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |

Table 4.1‑5 Table: Room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| RoomID | int | NN | Primary Key | Auto Increment Start With 1 |
| RoomNumber | Int | NN | Unique Key | Unique Room Number |
| RoomType | varchar(100) | NN | - | Type of the room (e.g., Single, Double) |
| Capacity | Int | NN | - | Maximum number of occupants allowed |
| Availability | varchar(10) | NN | - | Current status of the room |

# Stories and Scenario

## Story-1: Register new students and assign them rooms

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Hostel Manager,  I want to register new students and assign them rooms  So that their accommodation is efficiently managed. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | Efficient registration and room assignment are critical to ensure that all students have appropriate accommodation and that room allocation is optimized. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Student with Valid Information |
| Prerequisite | **:** | The Hostel Manager is logged in to the Hostel Management System. |
| Acceptance Criteria | **:** | **Given:**  The Hostel Manager is on the "Register Student" page.  **When:** The Hostel Manager enters valid student details, including name, student ID, contact information, and selects a room.  **Then:** The system successfully registers the student, assigns the selected room, and displays a confirmation message with the student’s registration number and room details. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Assigning a Room to a Student. |
| Prerequisite | **:** | The student is registered in the Hostel Management System. |
| Acceptance Criteria | **:** | **Given:** The Hostel Manager is on the "Room Assignment" page, and the student is already registered.  **When:** The Hostel Manager selects an available room and assigns it to the student.  **Then:** The system updates the room's status to occupied and records the assignment in both the student's profile and the room occupancy records. |

## Story-2: Manage maintenance requests efficiently

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Maintenance Manager,  I want to manage maintenance requests efficiently  So that any issues in the hostel are resolved promptly. |
| Priority | **:** | Medium |
| Estimate | **:** | M |
| Reason | **:** | Quick resolution of maintenance issues is essential for ensuring the comfort and safety of students in the hostel. |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Logging a Maintenance Request |
| Prerequisite | **:** | The student or staff member is logged in to the Hostel Management System. |
| Acceptance Criteria | **:** | **Given:** The user is on the "Maintenance Request" page.  **When:** The user submits a request with valid details, including the type of issue, room number, and description.  **Then:** The system logs the request, assigns it a priority level, and notifies the Maintenance Manager. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.2* | : | Updating the Status of a Maintenance Request |
| Prerequisite | **:** | The Maintenance Manager has received a maintenance request. |
| Acceptance Criteria | **:** | **Given:** The Maintenance Manager is on the "Maintenance Request" page.  **When:** The Maintenance Manager updates the status of the request (e.g., "In Progress," "Completed").  **Then:** The system records the update, sends a notification to the requester, and updates the request's history. |

## Story-3: View my room details and upcoming events

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As a Hostel Resident,  I want to view my room details and upcoming events  So that I can stay informed and manage my time effectively. |
| Priority | **:** | Medium |
| Estimate | **:** | S |
| Reason | **:** | Providing access to room details and events helps students feel more engaged and ensures they are aware of important information. |

### Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.1* | : | Viewing Room Details |
| Prerequisite | **:** | The student is logged in to the Hostel Management System. |
| Acceptance Criteria | **:** | **Given:** The student is on the "My Room" page.  **When:** The student views room details, including room number, amenities, and roommate information.  **Then:** The system displays the correct information and allows the student to request changes if needed (e.g., room change request). |

### Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.2* | : | Viewing Upcoming Hostel Events |
| Prerequisite | **:** | The student is logged in to the Hostel Management System. |
| Acceptance Criteria | **:** | **Given:** The student is on the "Events" page.  **When:** The student views the list of upcoming events.  **Then:** The system displays the events with details, and the student can RSVP or express interest in attending. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Hostel System | Test Designed by: | K. V. Gohel |
| Module Name: | **Login** | **Test Designed date:** | 09-09-2024 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 10-09-2024 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-condition**: **Web application should be accessible** | | | | |
| **Test Case ID** | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| **TC\_001** | Login to web application with valid credential | Functional | Login to Hostel management system web application through valid credential | TC\_001 |
| **TC\_002** | Login to web application with invalid credential | Functional | Login to Hostel management system web application through invalid credential | TC\_002 |
| **TC\_003** | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  [krishgohel173@gmail.com](mailto:Rbgondaliya@gmail.com) |  |
| 3 | Enter valid Password in Password field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and accepted | pass |  | Password: kvgohel |  |
| 4 | Enter valid captcha code in captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step required when human action validation perform | get captcha from image which is near by captcha field |  |
| 5 | Click on login button | User should login into site and navigated to dashboard | User navigated to dashboard and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with invalid Username and invalid Password | Should be display an error message enter wrong username or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and invalid Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 4 | Verify that User is not able to Login with blank Username or Password | Set required field validation message for Username and Password | Display an error of wrong username and password | Fail | Not performa a validation function fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link. | All listed control displayed properly on the page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| 7 | verify that forgot password link working properly | when click on forgot password load forgot passworg page | forgot password link not working | Fail |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml